Section III:

AMENDMENT UNDER 37 CFR §1.121 to the DRAWINGS

No amendments or changes to the Drawings are proposed.

Section IV:

AMENDMENT UNDER 37 CFR §1.121

REMARKS

Response to Previous Reply and Amendment

We appreciate the reconsideration by the Examiner and withdrawal of the objections to claims 2, 9, and 16; of the rejections under 35 U.S.C. §103(a) of claims 1-5, 7-12, 14-19 and 21 over Shefi in view of Müller; and of the rejections under 35 U.S.C. §103(a) of claims 6, 13, and 20 over Shefi in view of Müller in further view of Douceur.

Rejections under 35 U.S.C. §103(a)

Claims 1-5, 7-12, 14-19 and 21. Whereas Shefi was previously cited in the first Office Action, we will not repeat those arguments in this reply as a convenience to the Examiner. We respectfully maintain our arguments regarding Shefi's disclosure as set forth in our previous reply. In the new combination which relies upon Hattick in combination with Shefi, we agree that Shefi does not disclose a code exchanger as we have claimed. We respectfully disagree that Hattick's code exchanger disclosed at paragraphs 0017, 0019, 0023, and 0024, is the same as our claimed code exchanger.

We respectfully ask the examiner to consider our claim term (current amendments show):

> . . .
>
> a code exchanger configured to receive ~~for receiving~~ a One Time Pad [[pad]] value from said client device by said service security server upon request for initiation of a service session;
>
> . . .

Please note that our One Time Pad (OTP) value is being exchanged, and it is being sent in the direction *from* the client device *to* the server.

Hattick's Authentication Process (Fig. 3). We believe that Hattick does not exchange an OTP value *during authentication,* but instead exchanges *other values*. Here is why: If, one assumes that Hattick's "reader" corresponds to our "server", and if one assumes Hattick's "data carrier" corresponds to our client, then Figure 3 shows what is exchanged, and what is not exchanged, during authentication. In Hattick's third step (not counting the "Start" block), their "tag" or "data carrier" (corresponding to our "client") contains an OTP value, but the "reader" (corresponding to our server) does not have an OTP value, at least not yet. So, their "tag" or "data carrier" (corresponding to our "client") sends a "unique ID", a counter value, an encrypted challenge sequence, and an authentication value to their "reader" (corresponding to our server). Please note that the OTP value is none of these transmitted values – all of these transmitted values are something else.

When Hattick's "reader" (corresponding to our server) receives from their "tag" or "data carrier" (corresponding to our "client") the "unique ID", the counter value, the encrypted challenge sequence, and the authentication value, it must *generate* a OTP value from it. They denote the OTP value as **G(K, UID)** in their disclosure.

This interpretation of Fig. 3 is supported by their disclosure in ¶0024, where Hattick states [0024] (italicized notations and emphasis added by us):

> . . . the data carrier replies *[to the reader]* with its UID in plaintext, the increment counter value (i) in plaintext, the challenge sequence in cipher text . . ., and an authentication value (m). . . . Using the secret key (K) and the UID of the data carrier, the reader **generates** the unique one-time pad of the data carrier G(K, UID). (lines 17 - 24)

So, during authentication operation, we believe Hattick's system does not actually receive and OTP from a client, but instead an OTP is *generated* from other received values.

Hattick's OTP Programming Process (Fig. 2). Please also consider Hattick's Figure 2, in which an OTP value is exchanged, but in the reverse direction as we have claimed. During programming and initialization of their "tag" or "data carrier" (corresponding to our "client"), a "programmer" system is used. If one presumes that their "programmer" corresponds to our "server", then the comparison yields two significant differences between their process and ours. First, their *programming* process is not an "authentication" process because it is performed in a secure environment, according to Hattick's ¶0021, lines 18 - 19. Thus, being in a secure environment, no authentication is necessary. Second, and probably more significantly, their programming process sends the OTP value *from* their "programmer" (corresponding to our "server") *to* their "tag" or "data carrier" (corresponding to our "client"). This is the reverse direction of our claimed OTP value exchange *from* a client *to* a server.

Hattick's ¶0023 provides an interesting alternative embodiment where it discusses transmission of OTP values *to* and application server, but again, not *from* the "tag" or "data carrier" (corresponding to our "client"), but instead *from another server* (e.g. from a *key server*) (¶0023, lines 13 - 14). Still, this is not the same as transmission of an OTP value *from a client* to a server, nor is it part of an authentication process.

Please also consider that Hattick's Figure 2 illustrates this process in the same manner as we are proposing that it should be interpreted – the "tag" or "data carrier" (corresponding to our "client") sends the UID (not the OTP) value to the "programmer" (corresponding to our "server) in the third step, the programmer *generates* a OTP value (does not receive it from the tag) in the fourth step, and then the OTP is transferred from the "programmer" to the "tag" or "data carrier" (corresponding to our "client") (¶0022, lines 1 - 3), which is the reverse direction of OTP exchange that we have claimed (e.g. from the client to the server).

Hattick's Counter. Another point of difference with our claims with respect to Hattick's invention is their dependence on a counter, and the exchange of their counter value. Please refer to our previous reply remarks regarding Müller's counter, and our previous amendment specifying that a counter is not needed or used in our claimed invention.

Likewise, Hattick employs a counter as well, which they refer to as their "increment only counter", the value *(i)* is transmitted from their "tag" or "data carrier" (corresponding to our "client") to their "reader" (corresponding to our server) during their authentication process. We have specifically claimed no need for a counter value in our process, and our process is especially free of exchanging such a counter value.

If one were to remove Hattick's counter from their process in order to meet our claims, Hattick's method of generating a OTP value would be inoperative, whereas their counter value is required to complete the generation of the OTP value.

Thus, modifying Hattick to meet our claims would render Hattick inoperable, and unsatisfactory for its intended purpose, and there could be no motivation to make such a change. Where there is no motivation to make a change, the change is not obvious under 35 U.S.C. §103(a).

For these reasons, we believe that Shefi in view of Hattick fails to teach all of our claim elements, steps, and limitations, and that it would not have been obvious to modify Shefi and Hattick to meet our claims for lack of motivation and suggestion to make the necessary changes.

We respectfully ask for the Examiner to reconsider the rejections, and to allow Claims 1 - 5, 7 - 12, 14 - 19 and 21.

Claims 6, 13, and 20. In the Office Action, previously-cited Shefi was relied upon in view of newly-cited US published patent application to Hattick et al. ("Hattick") in further view of previously-cited Douceur. Whereas Shefi and Douceur were previously cited in the first Office Action, we will not repeat those arguments in this reply as a convenience to the Examiner. We respectfully maintain our arguments regarding Shefi's and Douceur's disclosures as set forth in our previous reply.

Claim 6 depends from Claim 1, Claim 13 depends from Claim 8, and Claim 20 depends from Claim 15. We believe Shefi in view of Hattick in further view of Douceur fails to disclose all of our claim elements, steps, and limitations, and that it would not have been obvious to modify Shefi, Hattick, and Douceur to meet our claims for lack of motivation and suggestion to make the necessary changes, as discussed in the paragraphs above.

We respectfully ask for the Examiner to reconsider the rejections, and allow Claims 6, 13, and 20.

Respectfully,

/ Robert Frantz /

Robert H. Frantz, Reg. No. 42,553
Registered U.S. Patent Agent, № 42,553
Tel: (405) 812-5613
Franklin Gray Patents, LLC